

## REMARKS

### Status of the Claims

- Claims 1-16 are pending in the Application.
- Claims 1-16 are rejected by Examiner.

### Specification Amendments

Paragraph [0040] is amended to include the assigned serial number and filing date of a reference application in the specification. Applicant submits that no new subject matter has been added as a result of the amendment.

### Claim Rejections Pursuant to 35 U.S.C. §102

Examiner has rejected Claims 1-16 under 35 U.S.C. §102(e) as being anticipated by U.S. Pat. No. 6,934,712 to Kiernan et al. Applicant respectfully traverse the §102(e) rejection.

Initially, Applicant notices that there is a subject matter scope difference between the elements of Claim 1 and Kiernan et al. Claim 1 is directed to a method for constructing an optimal representation for an input query, whereas the scope of the 35 U.S.C. §102(e) rejection uses multiple aspects of a much larger process that Kiernan et al. teaches.

Kiernan et al. in Figure 14 teaches receiving an input query and generating an intermediate representation which describes how the query is related to relational database tables (Figure 14, step 1400). The intermediate representation is rewritten to an equivalent form that may easily be translated to a SQL query. (Fig. 14, step 1402 and col. 14, lines 47-49). The intermediate representation is then translated to a SQL query and a tagger graph. The tagger graph, generated from the intermediate representation, describes how to generate an XML output from the results of a SQL query. (Fig. 14, step 1404 and col. 14, lines 52-60). The SQL query is then executed in a tagger component (see Fig. 2, item 240) using the tagger graph in order to generate an XML output of the SQL query. (Fig. 14, step 1406 and col. 14, lines 61-63).

Applicant notes that Kiernan et al. generates an intermediate representation from an input query and then generates a SQL query and a tagger graph from that representation. In

Kiernan et al, the intermediate representation is no longer modified after it is used to generate the SQL query and tagger graph in step 1404 of Figure 14. A tagger graph is defined in Kiernan et al. as:

The tagger graph is generated from the intermediate representation and, therefore, reflects the structure of the resultant XML document. (col. 7, lines 59-62).

Applicant notes that the tagger graph represents the structure of the XML output document. In addition, the tagger component (Fig. 2, item 240) accepts a SQL query as well as the tagger graph in order to produce the desired XML document from the relational database source. As stated in Kiernan et al.:

The tagger runtime 240 walks the tagger graph applying the operators of each node to the inputs at that node to produce the structured XML document as a query result, guided by the structure of the tagger graph. (col. 7 lines 61-64).

Claim 1 recites:

1. A method for constructing an optimal representation for an input query, the method comprising:
  - receiving the input query, wherein the input query is an intermediate language representation comprising nodes, each node having a respective node type;
  - examining the nodes in a left-depth first manner to identify node types for optimization;
  - tagging nodes corresponding to the identified node types;
  - moving upward to the next node until the intermediate language representation of the input query has been examined in its entirety;
  - searching from the top of the intermediate language representation for tagged nodes and identifying associated code patterns to be optimized; and
  - adjusting the identified code patterns with improved code patterns to form an optimal representation for the input query.

Applicant notes that the scope of Claim 1 is that of “constructing an optimal representation for an input query” as recited. The Examiner has cited (Page 3 of present Office Action) a portion of Kiernan et al. entitled “Tagger Operation” on col. 7 lines 52-64 to represent the Claim 1 element of “tagging nodes corresponding to the identified node

types”. Applicant respectfully disagrees that Kiernan et al. teaches this element of Claim 1. As noted above, a tagger graph is generated AFTER the intermediate representation is generated (See Figure 14). The tagger 240 operates when a SQL query and a tagger graph are input to the tagger 240. Claim 1 does not include a tagger, a tagger graph, or a SQL query in the steps to construct an optimal representation for an input query. In fact, there is no optimal intermediate representation in Kiernan et al. that is generated at all as a result of using a tagger graph. Thus, Applicant concludes that Kiernan et al. does not teach the Claim 1 element of “tagging nodes corresponding to the identified node types”.

Further, Applicant submits that whereas Kiernan et al. uses a tagger graph and a SQL query, both derived from a rewritten intermediate representation, and feeds them both into a tagger mechanism to produce an XML document from a relational database, Claim 1 recites steps to optimize an intermediate representation. The scope and resultant output of the two processes are different. The method of Claim 1 produces an optimized representation of an input query whereas the method of Kiernan et al. produces an XML document that was once a relational table. Applicant submits that the tagger graph of Kiernan et al. is not involved in an optimization of an intermediate representation of an input query as recited in Claim 1. The tagger graph of Kiernan et al. is generated from the rewritten intermediate representation of Kiernan et al. The teachings of Kiernan et al., which are beyond the scope of generating an optimal intermediate representation, cannot reasonably be used to reject Claim 1 because the scope of the two methods differ. Also, Kiernan et al. simply does not teach all of the steps involved in generating an optimal intermediate representation recited in Claim 1.

Similarly, the Claim 1 step of “moving upward to the next node until the intermediate language representation of the input query has been examined in its entirety” cannot relate to the tagger and tagger parse tree as indicated on page 3 of the present Office Action because they do not involve the optimization of the intermediate representation as recited in Claim 1. Therefore, col 9, lines 40-50 of Kiernan et al. which describe the tagger operations and tagger parse tree cannot be used to reject the step of “moving upward to the next node until the intermediate language representation of the input query has been examined in its entirety” because the Kiernan et al. tagger operation and parse tree are well beyond the scope of

optimizing an intermediate representation as recited in Claim 1. As noted above, the tagger graph of Kiernan et al. is generated AFTER the rewritten intermediate representation of Kiernan et al. is generated (See Figure 14 steps 1402 and 1404) and thus are not involved in an optimization of the intermediate representation.

Similarly, the Claim 1 step of “searching from the top of the intermediate language representation for tagged nodes and identifying associated code patterns to be optimized” cannot relate to the tagger graph as indicated on page 3 of the present Office Action because they do not involve the optimization of the intermediate representation as recited in Claim 1. Therefore, col. 10 lines 6-31 of Kiernan et al. which describe the tagger graph cannot be used to reject the step of “searching from the top of the intermediate language representation for tagged nodes and identifying associated code patterns to be optimized” because the Kiernan et al. tagger graph is generated from a rewritten intermediate representation as described by Kiernan et al. above and thus are well beyond the scope of optimizing an intermediate representation as recited in Claim 1.

Applicants respectfully submit that the elements of Claim 1 and the Kiernan et al. process of publishing relational data as XML by translating XML queries into queries against a relational database (See Abstract) are not similar in either general scope or specific elements. Specifically, Kiernan et al. fails to teach at least the Claim 1 steps of “tagging nodes corresponding to the identified node types, moving upward to the next node until the intermediate language representation of the input query has been examined in its entirety, and searching from the top of the intermediate language representation for tagged nodes and identifying associated code patterns to be optimized” in the context of optimizing an intermediate representation as recited in Claim 1.

Applicants note that independent Claims 1, 5, 6, and 8 have some similar elements which include generation or construction of an optimized intermediate language representation. Applicant submits the scope of these claims is not as broad as the scope of Kiernan et al. which uses the results of an intermediate representation to generate a tagger graph and a SQL query to subsequently use a tagger mechanism to publish relational data in XML form (Abstract).

Since Kiernan et al. fails to teach or even suggest that an optimized intermediate representation that is generated using the steps of “tagging nodes corresponding to the identified node types, moving upward to the next node until the intermediate language representation of the input query has been examined in its entirety, and searching from the top of the intermediate language representation for tagged nodes and identifying associated code patterns to be optimized”, as recited in the claims, then Kiernan et al. cannot anticipate independent Claims 1, 5, 6, and 8 and their respective dependent claims.

Concerning independent Claims 11, 15 and 16, Applicant submits that Kiernan et al. fails to teach the elements of “examining the nodes to inspect code patterns associated with respective node types, comparing the inspected code patterns using a pattern match algorithm to detect non-optimized code patterns, and adjusting one or more of the non-optimized code patterns and the inspected code patterns with improved code patterns to form an optimal representation for an input query” in the context or scope of constructing an optimal representation for an input query as recited in Claims 11, 15, and 16. For the reasons discussed above, Applicant submits that the teaching of Kiernan et al. which includes a tagger mechanism and a tagger graph exceeds the scope of the Claim 11, 15 and 16 which recite construction of an optimal representation for an input query. Since Kiernan et al. does not teach all of the elements of independent Claims 11, 15 and 16 and their respective dependent claims, then Kiernan et al. cannot anticipate Claims 11-16.

Applicant therefore respectfully requests withdrawal of the 35 USC §102(e) rejection on independent Claims 1, 5, 6, 8, 11, 15, and 16 and their related dependent claims because these claims patentably define over the cited art.

**DOCKET NO.:** MSFT-1744/303845.1  
**Application No.:** 10/601,480  
**Office Action Dated:** February 1, 2006

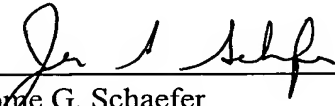
**PATENT**

**Conclusion**

Applicant respectfully requests reconsideration of all pending claims in light of the amendment and discussion above. Applicant respectfully submits that all pending claims patentably define over the cited art.

Respectfully Submitted,

Date: April 28, 2006

  
\_\_\_\_\_  
Jerome G. Schaefer  
Registration No. 50,800

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439